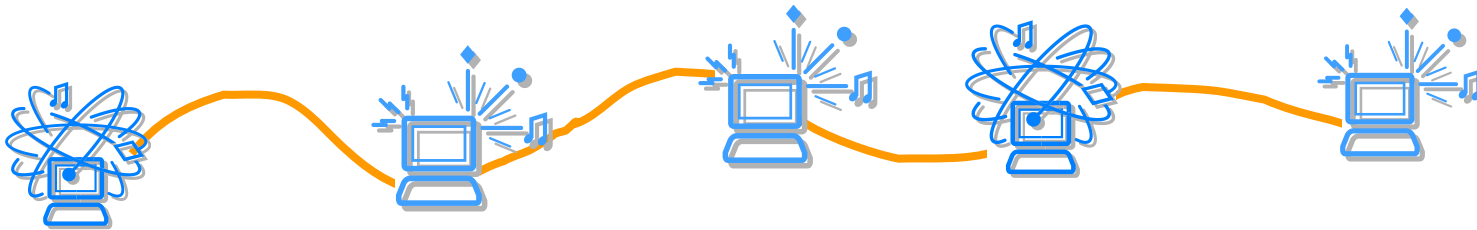




# OSPF - *Open Shortest Path First*

---



Redes de Comunicação de Dados

---



- RFC 2328 - OSPF Version 2  
<http://www.rfc-editor.org/rfc/rfc2328.txt>
- OSPF Design Guide, Cisco Systems  
[http://www.cisco.com/en/US/tech/tk365/technologies\\_white\\_paper09186a0080094e9e.shtml](http://www.cisco.com/en/US/tech/tk365/technologies_white_paper09186a0080094e9e.shtml)

# OSPF – Open Shortest Path First



- Fundamentos
- Áreas
- Comunicação entre routers OSPF
- Vizinhos, Adjacências e *Designated Routers*
- Bases de dados de LSAs: Sincronização e *Flooding*
- Rotas intra-área
- Rotas inter-área
- Rotas inter-AS
- Exemplos: rotas intra-área
- Exemplos: rotas inter-área



**Criação de um protocolo de *routing* interno, (IGP - *Internal Gateway Protocol*), dinâmico e “aberto” que possa ser implementado por qualquer fabricante, sem pagamento de licença, de modo a poder ser utilizado por todos...**

# Porquê um protocolo mais complexo?



- Os protocolos do tipo *vector-distance*, como o RIP, começam a “cair em desgraça” devido às suas limitações,
  - tempo de resposta a alterações, escalabilidade e quantidade de tráfego de “sinalização”
- OSPF não é o único protocolo do tipo *link state*
  - O protocolo IS-IS é parte do “*routing framework*” do OSI para o CLNP
    - É semelhante em desenho ao OSPF
    - Usa terminologia diferente
- Outros protocolos, alguns deles proprietários como o IGRP e EIGRP, são também uma alternativa possível.

# OSPF - *Open Shortest Path First*



- O OSPF é um protocolo do tipo *link state*, mais complexo que outros protocolos mais antigos, como o RIP, dado que:
  - A documentação é muito mais extensa
  - A gestão necessita de mais informação
  - A implementação necessita de mais código
- Então porquê desenhar um protocolo mais complexo?
  - Maior escalabilidade
  - Melhor controlo sobre o encaminhamento (*routing*)
  - Menos tráfego de “sinalização”
  - Calcula melhores rotas
  - O IETF tem vindo publicar extensões e melhoramentos

# Maiores vantagens do OSPF

---



- Não tem limites no número de saltos (*hop*)
- Suporta encaminhamento *classless*
- Tráfego menor dado as actualizações dos caminhos serem enviadas com intervalos muito grandes ou quando há uma mudança na topologia
- Convergência rápida
- Não cria *loops*
- Mais rápido a reagir às mudanças na rede
- Melhor balanceamento de carga
- Definição lógica de áreas permitindo “dividir para reinar”
- Marcação de rotas externas
- Autenticação

# OSPF e algoritmos de *link-state routing*



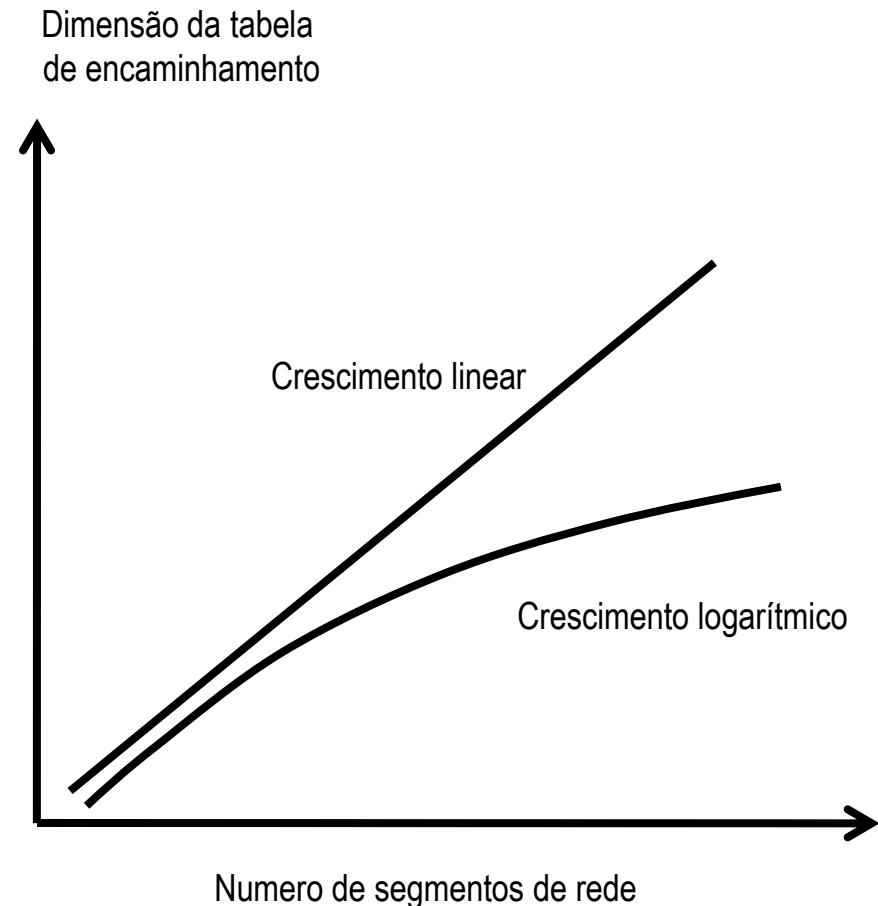
- Cada *router* constrói um “mapa” com a topologia da área. Para isso trocam mensagens (*Link State Update*) de cada um para todos os outros anunciando a quem estão ligados. Usam, sempre que possível, *multicast*.
- Cada *router* calcula, utilizando o algoritmo de Dijkstra (SPF), o caminho mais curto dele para todos os outros *routers* da área.
- A tabela de *routing* inclui a informação resultante do cálculo e ainda a informação proveniente dos *routers* que fazem fronteira com outras áreas.
- Os *routers* vigiam-se constantemente enviando mensagens (Hello) para verificarem se os outros continuam vivos. Se assim não for avisam os restantes *routers* da área e recalculam os melhores caminhos.



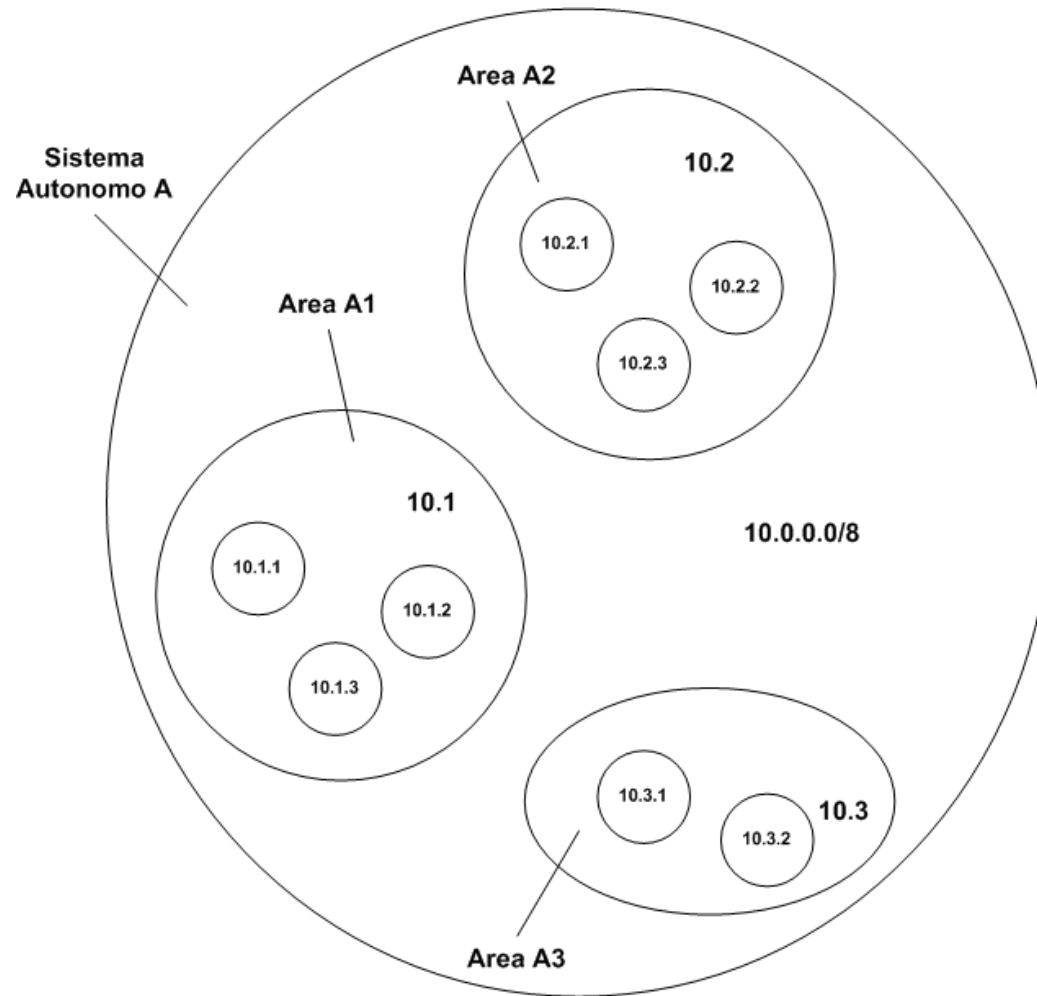
# Sistemas autónomos e áreas: Escalabilidade



- Conhecimento de todas as rotas implica:
  - Grande capacidade de memória
  - Troca de informação sobre todas as rotas
  - Troca de informação entre todos os routers
- A complexidade computacional cresce linearmente com a dimensão da rede
- Solução:
  - Dividir a rede em hierarquias para reduzir complexidade
  - **Utilizar algoritmos de encaminhamento diferentes para cada situação**



# Sistemas autónomos e áreas: Hierarquias



# Sistemas autónomos e áreas



- Sistema autónomo (AS), no contexto do OSPF, é um grupo de *routers* que trocam informação de encaminhamento entre eles através de um protocolo de encaminhamento comum.
- É constituído por um grupos de redes que pode ser dividido em vários grupos mais pequenos.
- Estes grupos são designados por áreas.
  - Um sistema autónomo pode ser composto por uma ou mais áreas.
  - A divisão em áreas tem a vantagem da topologia de cada área ser escondida das restantes áreas do sistema autónomo, da eventual instabilidade de uma área não afectar as restantes e permite ainda que os *routers* envolvidos necessitem de menos memória, dado que sendo as área mais pequenas o número de *routers* e de redes em cada uma é menor e as rotas para as outras áreas são sumarizadas.



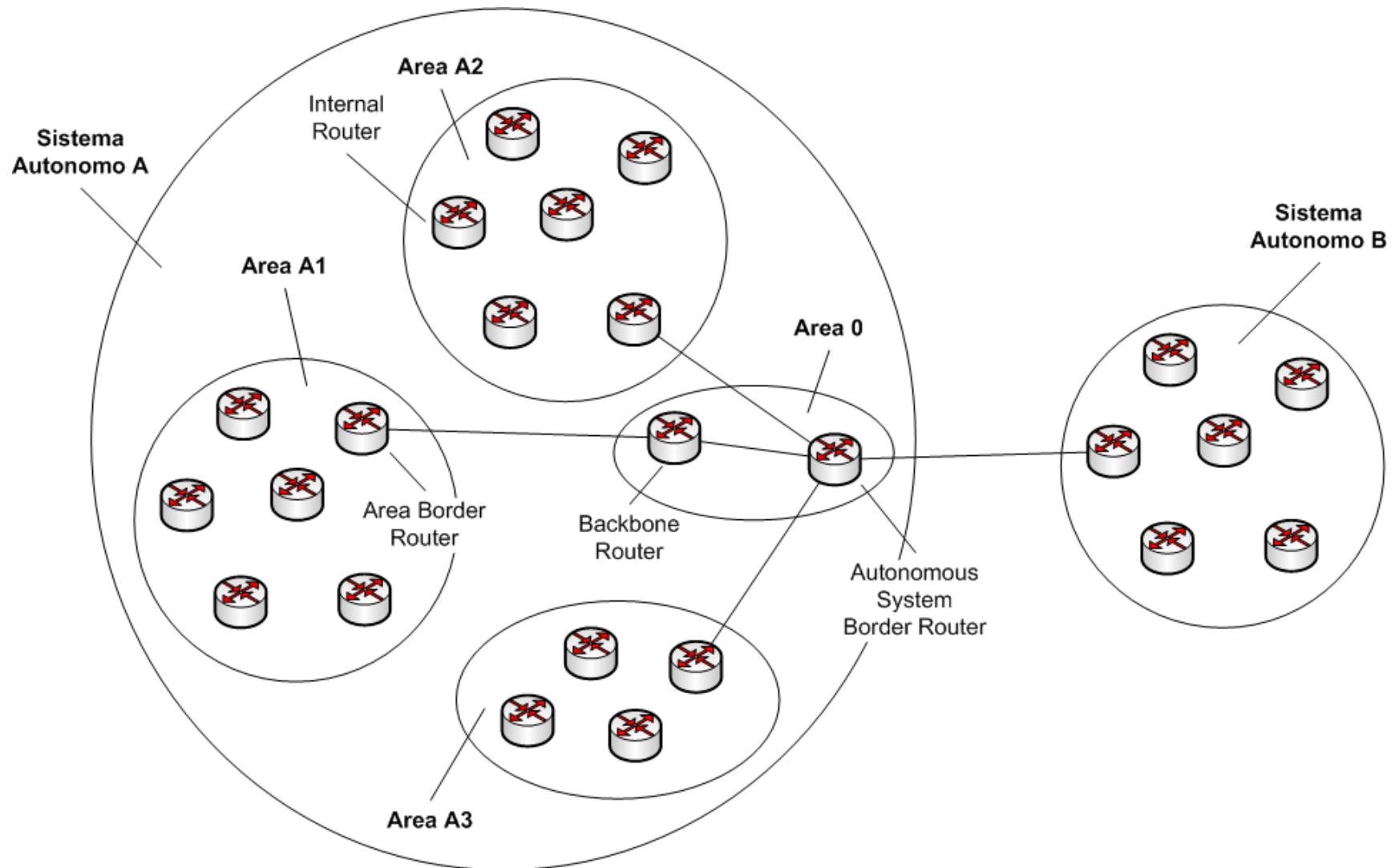
- Um sistema autónomo pode ser constituído (dividido) por uma ou mais áreas
- Só as máquinas de uma área é que trocam mensagens de descrição de base de dados entre si e têm o conhecimento da topologia dessa área.
- A topologia resultante forma uma árvore com dois níveis em que a raiz é a área 0 (*backbone*) e as folhas (segundo nível) as restantes áreas, evitando *loops* entre áreas.
- Vantagem da divisão em áreas:
  - Limita em cada *router*, o tamanho do grafo da topologia e o processamento necessário para calcular os melhores caminhos
  - Limita o número de ligações que podem mudar de estado provocando novo cálculo dos melhores caminhos nos *routers*
- Área ID é um número a 32 bits que identifica a área.

# A área de *backbone*



- A área de *backbone* interliga todas as outras áreas de um AS
  - Se houver problemas na área 0 todas as redes do AS sofrem
- Tem como principal função eliminar os problemas do algoritmo de vectores de distância, nomeadamente *loops*
- Todas as rotas inter-área são distribuídas pela área de *backbone*
- Todas as áreas têm ligações directas à área 0
  - Caso não exista uma ligação física directa entre uma área e a área 0, uma ligação virtual é estabelecida através de uma (única) outra área com ligação directa ao *backbone*

# Tipos de *routers*



# Tipos de *routers*



- *Internal Router*
  - Tem ligações apenas a *routers* da mesma área
- *Area Border Router (ABR)*
  - Tem ligações a *routers* de outra área (*backbone*), sendo o responsável pela troca de informações de *routing* entre áreas
  - Cada ABR numa área sumariza para a área o custo para todas as redes externas à área
  - Depois de ser calculada a árvore SPF para a área, os caminhos para os destinos inter-área (exteriores à área) são calculados examinando os sumários dos ABR
- *Autonomous System Border Router (ASBR)*
  - Tem ligações a *routers* de outros Sistemas Autónomos
  - Também pode executar outros protocolos de *routing* (IGP ou EGP - RIP, EIGRP, BGP)
- *Backbone Router*
  - Tem pelo menos uma interface que executa o OSPF na área 0



- *Router ID* é um **número de 32 bit** atribuído para identificação de cada **router OSPF** e que o identifica dentro do AS
- A norma aconselha a definir o *router ID* como o **menor endereço IP das interfaces do router**
  - Verifica-se que algumas implementações optaram pelo maior endereço IP como, por exemplo, a Cisco
- O *router ID* pode ser, por esta ordem de prioridade:
  1. Configurado **manualmente**,
  2. Obtido a partir dos endereços IP das **interfaces loopback**, ou
  3. Obtido a partir dos endereços IP das **interfaces físicas**,



# Interfaces de *loopback*



- Ao contrário das interfaces físicas, as quais se podem activar e desactivar por muitas razões, as interfaces *loopback* só são activadas ou desactivadas administrativamente pelo que o *router* ID se manterá estável.
- Se o *router* possuir pelo menos uma interface virtual ele irá escolher o seu *router* ID de entre os endereços das interfaces *loopback* em detrimento de qualquer das outras interfaces físicas.
- Estas interfaces são anunciadas nos router-LSA (LSA – *Link State Advertisement*) como caminhos simples para máquinas (*single host routes*), cujo endereço IP é o da interface

# Encaminhamento hierárquico



- **Encaminhamento intra-área**
  - Cada *router* tem toda a informação necessária para construir a sua tabela de *routing*. A tabela de *routing* é construída a partir da ***link state database*** (“mapa da rede”) da respectiva área. O *router* utiliza o algoritmo de Dijkstra para calcular a árvore dos caminhos mais curtos até aos vários destinos dentro da área.
- **Encaminhamento inter-área**
  - Os ABR das várias áreas trocam através do *backbone* a informação necessária para que se possa aceder a todas as redes do AS. Esta informação trocada pelos ABR é do tipo *distance-vector*
- **Encaminhamento inter-AS**
  - Os ASBR anunciam as rotas para as redes exteriores ao AS. Esta informação enviada pelos ASBR é do tipo *distance-vector*

# Vizinhanças e Adjacências

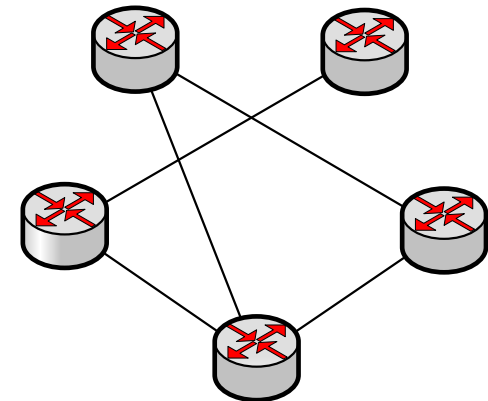
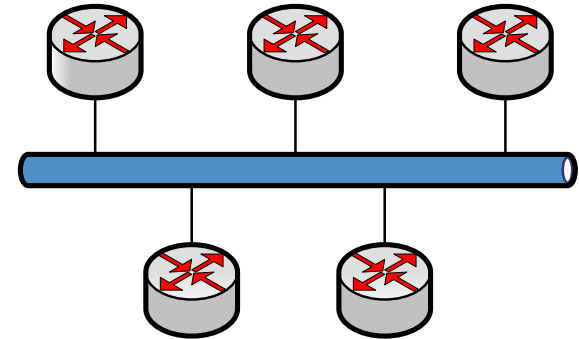


- *Routers* vizinhos (neighboring routers) são *routers* que têm interfaces numa rede comum (comunicam a nível 3 directamente). Se dois *routers* podem comunicar directamente entre si então podem estabelecer relações de vizinhança.
  - A relação de vizinhança é estabelecida e mantida pelo protocolo Hello
- Dois *routers* só serão vizinhos se estiverem de acordo nos seguintes pontos:
  - Identificação da área (area-id),
  - autenticação (authentication), e
  - intervalos de Hello e de Dead (intervalo máximo sem mensagens de Hello).
  - *Stub area flag*
- Adjacência (adjacency) é uma relação estabelecida entre dois *routers* vizinhos com o objectivo de trocar informação de *routing* (sincronizar as bases de dados).
  - As bases de dados entre *routers* adjacentes tendem a ser iguais se estes pertencerem à mesma área.
  - Nem todos os *routers* vizinhos são adjacentes.

# Tipos de rede



- O OSPF foi implementado de modo a suportar diferentes tipos de redes:
  - **Redes ponto-a-ponto**
  - **Redes de difusão de acesso múltiplo (BMA)**
    - O canal de comunicação é partilhado
      - e.g., Ethernet, TokenRing.
    - *Broadcast Medium Access*
  - **Redes de não difusão de acesso múltiplo (NBMA)**
    - O canal de comunicação é dedicado
      - e.g., X.25, FrameRelay
    - *Non-Broadcast Medium Access*
  - **Redes ponto-multiponto**
    - e.g., ATM



# Designated Router (DR)



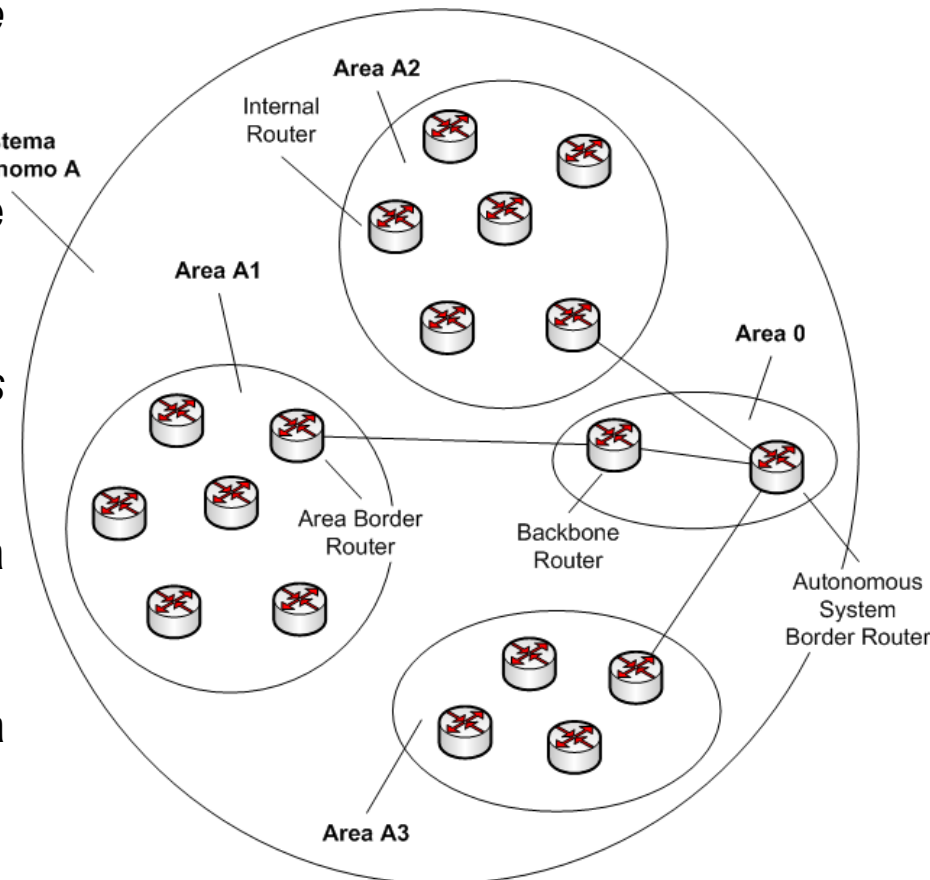
- Cada rede BMA ou NBMA que tenha pelo menos dois *routers* elege um *designated router* (nos equipamentos Cisco são eleitos DR mesmo que não haja vizinhos no entanto, neste caso, não geram LSA tipo 2).
- Os *designated routers* geram network-LSA (tipo 2) das redes BMA (Broadcast Multiple Access) e NBMA (Non BMA)
- **O conceito de *designated router* permite a redução no número de adjacências entre *routers* requeridas em redes BMA ou NBMA**
  - **Reduz a quantidade de tráfego relacionado com o OSPF**
- Todos os *routers* vizinhos são adjacentes do *designated router* e do *backup designated router* da rede
- Redes ponto-a-ponto têm adjacências entre os *routers* mas não têm *designated routers*

# Link State Advertisements (LSA)



Informação trocada entre os *routers* de cada área com o objectivo (final) de criarem as suas *link state database* e posteriormente as suas *tabelas de routing*.

- **Type 1 – Router LSA:** O *router* descreve as redes a que está ligado e respectivos custos
- **Type 2 – Network LSA:** O DR descreve os *routers* ligados a uma dada rede
- **Type 3 – Summary LSA:** Enviados pelos ABR de cada área para descrever as redes desta
- **Type 4 – AS Summary LSA:** O ABR descreve a localização dos ASBR
- **Type 5 – AS External LSA:** O ASBR descreve o sumário de destinos exteriores para os quais é Gateway



# Base de dados das ligações (*link state database*)



- Colecção dos router LSA, network LSA e summary LSA com origem na área. Esta informação só é propagada dentro da área onde têm origem.
- A lista de AS external LSA também é considerada como fazendo parte de base de dados de cada área.
- A partir dos LSA recebidos cada *router* numa área constrói a sua base de dados da topologia da rede (*link state database*) ficando a conhecer o “mapa” da sua área.
  - As *link state databases* dos vários *routers* de uma área são iguais entre si mas são diferentes das de outras áreas, dado que a informação proveniente de fora da área vem num formato vector-distance e sumariada na maioria das vezes
  - A *link state database* é composta por:
    - *Router-link state advertisements*, utilizado pelo algoritmo de Dijkstra
    - *Network-link state advertisements*, utilizado pelo algoritmo de Dijkstra
    - *Summary-link state advertisements*
    - *AS-external-link state advertisements*

# Encaminhamento intra-áreas



- A partir da base de dados da topologia (*link-state database*) da rede o *router* constrói uma árvore com os caminhos mais curtos com ele próprio na raíz. Utiliza o algoritmo de Dijkstra.
- A árvore dos caminhos mais curtos indica o caminho mais curto para cada destino dentro da área.
- Todos os *routers* correm exactamente o mesmo algoritmo, em paralelo.
- É crucial que todos os *routers* numa área possuam base de dados da topologia (*link-state database*) idênticas (sincronizadas).
- Caminhos derivados de informação de encaminhamento externo à área aparecem como folhas da árvore. São anunciados através do Sistema Autónomo mas são mantidos separados da informação de *link state*.



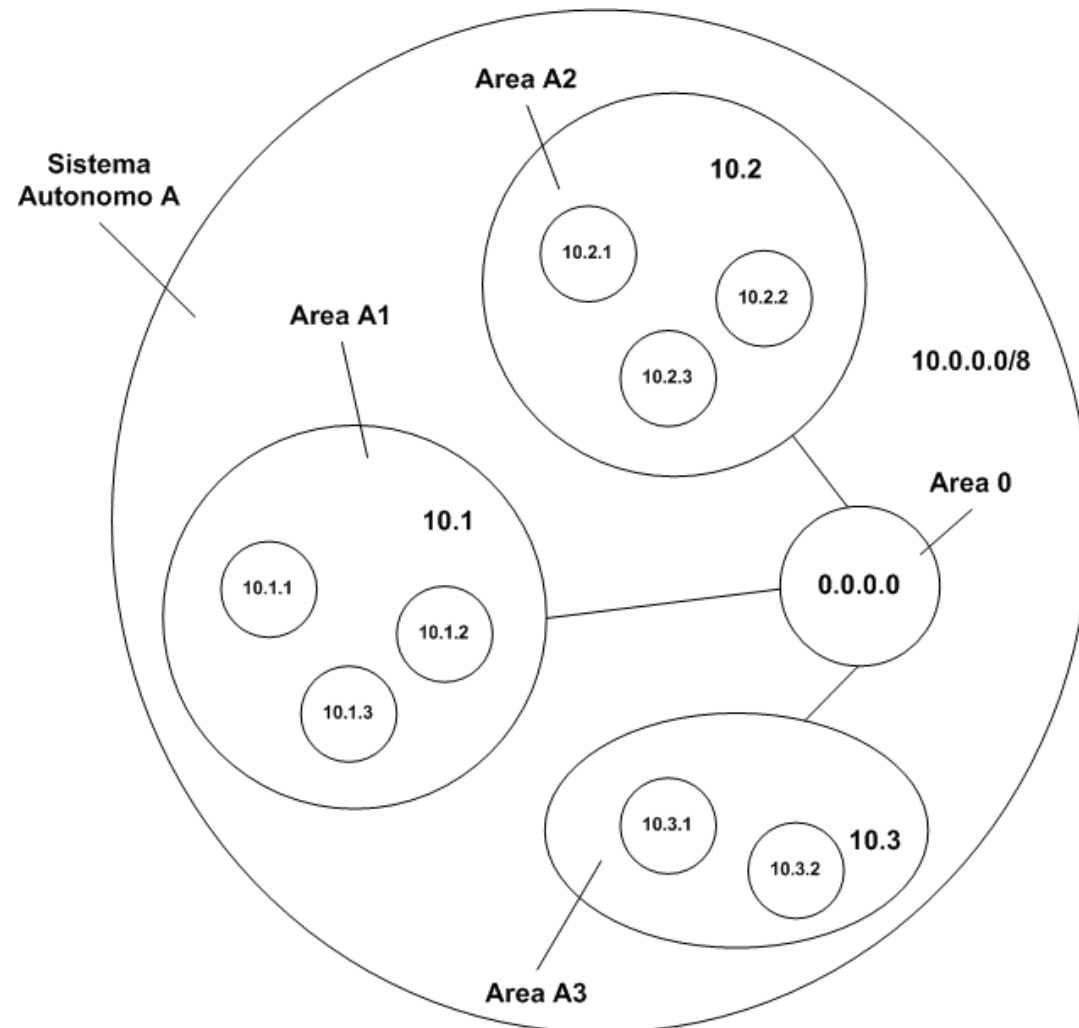


- As áreas anunciam sumário das suas rotas através do algoritmo de vectores de distâncias
- Para evitar os problemas associados a este tipo de algoritmos (*loops*) foi criada a área de *backbone*, deste modo todas as áreas comunicam através de uma única área central com a qual os algoritmos do tipo vector de distâncias lidam sem problemas de criarem *loops*
- Para permitir que todas as áreas, mesmo as mais remotas, se possam ligar à área de *backbone* foram criadas ligações virtuais que permitem utilizar uma área como um túnel entre o *backbone* e outra área.

# Encaminhamento inter-áreas: Área de *backbone*



- A área 0, ou 0.0.0.0, é a área de *backbone* e, se um domínio OSPF existe, tem de existir sempre esta área com este ID
- É a área que interliga todas as outras áreas de um AS
  - Se houver problemas na área 0 todas as redes das outras áreas do AS sofrem.
  - A utilização de uma área de *backbone* evita *loops* entre áreas.



# Encaminhamento diferenciado

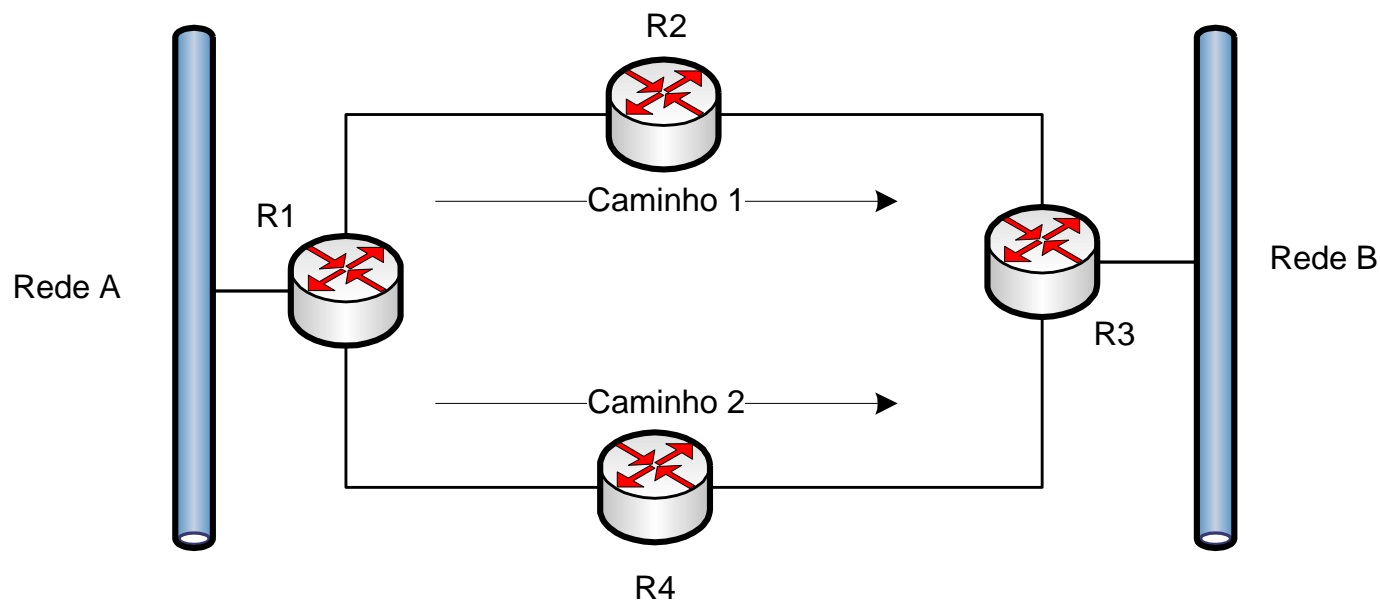


- O OSPF suporta várias métricas. A usada por omissão é o débito das ligações. O custo é calculado com  $1 \times 10^8 / \text{débito\_da\_ligação em bps}$ . O custo é adimensional.
- Se só for usado apenas uma métrica, o TOS (*Type of Service*) não é suportado
- Se forem utilizadas várias métricas, o TOS é suportado em opção dando origem a tabelas de *routing* separadas para cada uma das combinações possíveis dos três bits do TOS
  - *delay, throughput, reliability*
- Exemplo:
  - Se os bits do TOS do IP especificarem atraso reduzido, baixo débito e elevada fiabilidade o OSPF calcula caminhos para todos os destinos baseado na indicação do TOS.

# Encaminhamento: Balanceamento de tráfego



Balanceamento de carga por múltiplos caminhos



**O OSPF suporta caminhos com custos e tipos iguais.**



- **Aprendizagem da tabela de encaminhamento**
  - Cada *router* testa as suas ligações periodicamente (10 s) com outros *routers* vizinhos a que está ligado (**protocolo Hello**)
  - Cada *router* reconstrói a sua base de dados de ligações periodicamente (cada 30 m), através da sincronização com as bases de dados dos *routers* adjacentes (**protocolo Exchange**)
  - Cada *router* calcula o melhor caminho para cada destino utilizando o algoritmo **Shortest Path First – SPF (Dijkstra)**
  - Inclui na tabela de *routing* os melhores caminhos calculados
  - Anuncia as eventuais alterações que detecte nas ligações (**protocolo flooding**)
- **Encaminhamento**
  - Cada *router* encaminha pacotes para um destino enviando para o próximo *router* (de acordo com a tabela de *routing* )

# Passos para aprendizagem de rotas

---



- Definição de vizinhanças, adjacências e *Designated Routers*
  - Mensagens de Hello
- Inicialização da base de dados de ligações
  - Descrição da BD de ligações: Mensagens de Database Description
  - Solicitação das ligações desconhecidas (diferença): Mensagens de Link State Request
  - Envio da informação sobre as ligações desconhecidas: Mensagens de Link State Update
- Actualização da base de dados de ligações (**flooding**)
  - Envio da informação sobre alterações: Mensagens de Link State Update
  - Confirmação da recepção: Mensagens de Link State Acknowledge



- Escalabilidade
- Sistemas autónomos, áreas
- Vizinhanças e adjacências
- Tipos de LSA e bases de dados de ligações
- Tipos de rede: BMA, NBMA e ponto-multiponto
  - *Designated router*
- Encaminhamento hierárquico
  - Inter-área: Algoritmo de vectores de distâncias e a área de *backbone*
  - Intra-área: Algoritmo de estado da ligação
- Encaminhamento: ToS e balanceamento